Improvements in TRANSPORT

David C. Carey

April 14, 1972

I. INTRODUCTION

A simple charged particle beam line will typically be comprised of quadrupoles, bending magnets with or without pole face rotations, and drift spaces. Given an initial beam one often wishes to vary certain parameters to fit elements of a transfer or beam matrix. The program TRANSPORT[1] allows one to do these things accurately and with reasonable efficiency. However when one wishes to calculate something slightly more complex, such as second order, misalignments, sequential fitting, or beams whose centroid does not coincide with that of the beam line, one has, in the past, encountered certain errors, inefficiencies, or inconvenient features of the program.

We have therefore substantially rewritten the program to eliminate such difficulties. Below we explain in detail the changes made and the advantages thus gained. It will be assumed that the reader is familiar with the old version of TRANSPORT.

II. UPDATES AND CALCULATION OF MATRICES

A. The Transfer Matrices R1 and R_2

The first order effect of a beam line can be represented by a transfer matrix R_1. The beam matrix at any point may then be calculated from the initial beam matrix and the transfer matrix up to the point in question, by the equation $\sigma = R_1 \sigma_o R_1^T$, where σ is the beam matrix at the point of interest and σ_o is the original beam matrix. There are, however, certain operations in TRANSPORT which cause a change in the beam envelope, but which have no effect on the transfer matrix. Such operations, along with the type code used to represent them are:

1. rms addition to the beam
7. shift of beam centroid
8. misalignment
21. stray field

At any point where such an operation occurs, it is necessary for the program to "update" the beam. That is, to store values of the beam matrix elements at this point and reinitialize the R_1 matrix, so that the beam matrix at later points is calculated in terms of this new "initial" beam matrix and the reinitialized R_1 matrix. The R_1 matrix could also be updated explicitly via a 6. 0. 1. element.

At times, however, one is interested in knowing the elements of a transfer matrix from some point in a beam line, such as the beginning, previous to such an "update". For this reason, the old version of TRANSPORT allowed one to introduce a completely independent transfer matrix R_2 which was not updated by any of the above elements. It could be initialized at any point desired and was calculated only when initialized at some previous location.

A disadvantage of this approach was that, once R_2 had been introduced, the program needed to take the time to calculate both R_1 and R_2. When passing through an element with transfer matrix R, the matrices R_1 and R_2 respectively were changed as follows:

$$R \times R_1 \rightarrow R_1$$
$$R \times R_2 \rightarrow R_2$$

If one were doing fitting, then one had the possibility of fitting elements of either R_1 or R_2, and therefore the program needed to calculate partial derivatives of both matrices with respect to any varied quantity. Since the matrices R_1 and R_2 are independent their partial derivatives must be calculated independently as follows:

$$R \times \frac{\partial R1}{\partial v} + \frac{\partial R}{\partial v} \times R1 \rightarrow \frac{\partial R1}{\partial v}$$

$$R \times \frac{\partial R2}{\partial v} + \frac{\partial R}{\partial v} \times R_2 \rightarrow \frac{\partial R2}{\partial v}$$

where v is a quantity which is being varied. Since varied quantities may be coupled both terms on the left side of one of these expressions may at times be nonzero.

A further difficulty was that there were no second order terms carried with R_2, so in such cases one had to eliminate all updates and refer again to R_1. Both in the extra time used to carry two separate matrices, and in the need to switch back and forth between the two to determine the transfer matrix for the entire beam line, this arrangement proved to be rather inconvenient.

In the new version of TRANSPORT the matrices R_1 and R_2 are given much more clearly defined and distinguished functions, but are not as independent. The matrix R_1 is always the one which represents the entire beam line. The only element which updates R_1 is a 6. 0. 1. element, i.e., an explicit update of the R_1 matrix. The transfer matrix used to calculate the beam matrix is R_2, and therefore the elements mentioned above, those that affect the beam directly, update only the R_2 matrix. The matrix R_2 is also updated by a 6. 0. 2. card. Both R_1 and R_2 matrices are initiated at the beginning of the beam line. Now, as one progresses down a beam line only the R_2 matrix and its partial derivatives are accumulated. Until an element occurs which would update R_2, the matrix R_1 and its partial derivatives are assumed equal to R_2 and its partial derivatives. When R_2 is updated, the matrix R_1 and its partial derivatives are calculated at the location of the update and saved by the program. If at some later point in the beam line a printing of the R_1 matrix is requested by a 13. 4. card, the R_1 matrix at that point is calculated as the product of the R_2 matrix at that point and the R_1 matrix at the last R_2 update. If no R_2 updates have occurred the R_1 matrix is taken as equal to the R_2 matrix. If the R_1 matrix is updated, the R_2 is also, automatically. The R_2 matrix now also carries second order elements. If one wishes to fit an element of the R_1 matrix, the matrix element and its partial derivatives are calculated similarly. However, in this case, the program calculates the value and partial derivatives only of the matrix element being fit.

The R_2 matrix and its partial derivatives are then calculated continually as one progresses down the beam line; the R_1 matrix elements and their partial derivatives are calculated only as needed. The need, in all cases, to accumulate only one transfer matrix as one progresses down the beam line results in a substantial improvement in efficiency. The ability to refer to a single matrix, R_1, as being that which represents the entire beam line is more convenient and eliminates confusion.

B. Constraints on The Beam Matrix

Previously, a constraint on the beam matrix would cause an update on the R_1 matrix. A beam matrix constraint caused no change in the beam at its point of occurrence, so there was no intrinsic need for such an update. It occurred simply because the portion of the program where the update was done was the only part where the partial derivatives of the beam matrix with respect to the varied parameters were calculated.

In the present version of the program, no updates occur when a constraint is placed on the beam. The partial derivatives of only the elements being constrained are calculated. This method is more efficient than a calculation of the entire matrix of partial derivatives, providing fewer than six beam matrix elements are being constrained at the same point.

C. Loss of Beam Centroid on Update

Several elements in TRANSPORT cause a shift in the beam centroid, so that it no longer coincides with the axis of the beam line. Such elements are:

7. Shift in the beam centroid
8. Known misalignment
17. Second order
21. Known stray field

In the old version of TRANSPORT, whenever such an element was included in a beam line and at some later point an update was made, the displacement of the beam centroid was reset equal to zero. This difficulty is not present in the SLAC version, but appeared when the program was transferred to the PDP-10. There are a number of important cases where one would want to include such a combination of elements. Some examples are as follows.

A beam line may have a number of elements in it with known misalignments. Each misalignment will cause a shift in the beam centroid. One is interested in knowing the cumulative effects of all the misalignments specified. With the above mentioned defect one obtains only the effect of the last misalignment. Since one never sees the beam centroid set to zero, it will not be obvious that this is happening and a wrong answer will result.

The effect a bending magnet whose field is mis-set by a certain amount may be of interest.[2] One cannot simply change the field directly and run, since doing so will change the axis of the beam line and TRANSPORT represents the effect of the beam relative to this axis. One can simulate the altered field by giving the beam a non-zero fractional deviation of its central momentum from the design momentum of the beam line, at the beginning of the magnet. At the end one again shifts the fractional deviation in central momentum back to zero. The idea is that the centroid of the beam will be shifted in spatial and angular coordinates as it traverses the magnet due to the altered momentum. The alteration in momentum is removed while the spatial displacement persists. To effect this procedure one uses a 7. type code which is a centroid shift, and causes an update. The effect of the update was to set all centroid displacements equal to zero making the above procedure impossible.

A constraint on the beam matrix could be made in a beam where some previous element has caused a shift in the beam centroid.[3] If two such constraints were made consecutively, the second would be working with spurious beam parameters rendered irrelevant by the unfortunate effect of the update. If such constraints were at the end of the beam line, there would be no subsequent printing of the beam matrix and no indication of what had happened. The program would simply go through some procedure and produce incomprehensible results.

The new version of TRANSPORT is constructed so that the centroid displacement is retained along with its partial derivatives with respect to any varied parameters. The effect of the 7. type code should now be specified as a <u>shift</u> in the beam centroid, the parameters on the card being added to the previous coordinates of the centroid.

III. SECOND ORDER CALCULATIONS

A. Second Order with Beam Correlations or Displacements

If one specifies a ray at the beginning of a beam line via a ray vector x^o, then in a first order calculation the ray vector at a later point x^1 has its coordinates given by:

$$x_i^1 = \sum_j R_{ij} x^o{}_j.$$

If, by inserting a 17. card into a TRANSPORT deck one includes the effect of a second order matrix where the coordinates of x^1 are given by:

$$x_i^1 = \sum_j R_{ij} x^o{}_j + \sum_{jk} T_{ijk}\, x^o{}_j x^o{}_k.$$

If one regards the elements of the beam matrix σ as being the second moments of a distribtuion, then it is possible to include second order effects in the calculation of the beam matrix and obtain an estimate of the beam envelope size at any point in the beam line.

The old version of TRANSPORT did all of the above, but based on the assumption that the initial ellipse was upright and on axis. Thus, there was no provision for including in a second order calculation the effects of correlations or centroid shifts as would be represented by type codes 12. and 7. respectively. A method of including these effects has been worked out and will be discussed in another report. This method has been incorporated in TRANSPORT.

Sometimes a centroid shift element is used to do single ray tracing. One can now include second order effects in this procedure.

B. Specification of Moments

In the earliest version of TRANSPORT which contained second order, one needed to specify the relative strengths of the moments of the beam distribution. The program then used these numbers to calculate a matrix of fourth moments from the elements of the beam matrix. The second moments of the beam distribution at any point in the beam line were then calculated from the initial second and fourth moments via the first and second order transfer matrices.

In specifying the relative moments the numbers 1. 0. 3. were used most often and came to be standardized. In later versions the program was changed to supply these numbers automatically.

The new version of TRANSPORT calculates the new second moments directly from the old second moments and the first and second order transfer matrices. It is therefore no longer appropriate to retain a moment specification in the data, and it is correspondingly not printed. Since the old moment specification was based on the assumption of a gaussian distribution the program now prints directly the words "gaussian distribution". If any user has decks of data where the moments are specified he should change them as the extra numbers, in some cases, may be interpreted to mean something else.

C. <u>Speed of Execution</u>

Previously the second order transfer matrix was handled by incorporating it into an expanded first order matrix. The dimensions of the new R matrix were 42 x 42 and its entries were determined as shown.

$$\left(\begin{array}{c|c} R & T \\ \hline O & R \times R \end{array}\right)$$

The upper left 6 x 6 submatrix is the first order transfer matrix. The upper right part contains a symmetrized T matrix and its dimensions are 6 x 36 or effectively 6 x 6 x 6, the last two indices being combined into a single index. The lower right submatrix is given by an outer product of the first order R matrix with itself. The remainder of the matrix is zero. This procedure of representing R and T matrices in a single matrix allows one to accumulate transfer matrices in second order exactly as was done in first order. The expanded total R matrix is just the product of the individual expanded R matrices.

Unfortunately the calculation of the products of the matrices written as R·R is time consuming and yields no new information. The expanded R matrix method has been replaced by one which represents the R and T matrices separately as 6 x 6 and 6 x 6 x 6 arrays respectively. Then if R^a, T^a and R^b, T^b are individual R and T matrices, the elements of the cumulative R and T matrices are calculated by the equations:

$$R_{ij} = \sum_{\ell} R^b_{i\ell} R^a_{\ell j}$$

$$T_{ijk} = \sum_{\ell} R^b_{i\ell} T^a_{\ell jk} + \sum_{\ell} T^b_{i\ell m} R^a_{\ell j} R^a_{mk}$$

where the T matrix represented here is the symmetrized matrix mentioned above.

This new method of matrix accumulation and the direct calculation of the beam matrix from the original beam matrix both result in a substantial improvement in computing efficiency.

In many cases the time required to perform a second order run of a beam line will be only 1/40 of that previously needed.

D. Bending Magnet Transfer Matrix

As the bending angle of a bending magnet approached zero, the expressions used for the second order transfer matrix elements in the old version of TRANSPORT became indeterminant, i.e., took the form 0/0. Therefore the program was written so that, if the bend angle was less than ten milliradians, the second order matrix elements were all set equal to zero. For a single bending magnet with a perfectly linear field this is not a bad approximation. However, the NAL main ring bends a beam 360° by a series of magnets, each of which has a bend angle of less than 10 mr. In addition one may have bending magnets with nonlinear fields. Such nonlinearities can have a significant effect on the profile of the beam. In order to determine this effect or to correct for it, one must be able to calculate the contribution of the field non-linearities to the second order transfer matrices.

The expressions for all the second order transfer matrix elements for a bending magnet have been reformulated to avoid the above difficulty. Now, when the bend angle goes to zero, all matrix elements approach either a definite non-zero limit or vanish. There is now therefore no restriction on the bending angle for which such matrix elements may be evaluated.

IV. MISALIGNMENTS

A. Incompatability with Rotation Element & Sextupoles

In order to calculate the effect of misalignments in a beam line, two different matrices are needed. One is the ordinary first order transfer matrix R, and the other is a matrix giving the shift in the central orbit of the misaligned section due to the misalignment parameters. The latter matrix is calculated from a matrix giving the spatial transformation of coordinates between the beginning and end of the misaligned section and the components of a vector spanning the misaligned section. These quantities are determined for an entire misaligned section by accumulating the expressions for all the individual elements of a section which have a first order transfer matrix.

In the earlier version of TRANSPORT, such terms were not given for either the beam line rotation element (20.) or for a sextupole (18.). Any misalignment calculation where the misaligned section contained such an element was therefore necessarily incorrect. These expressions have now been included in the program and misaligned sections can contain both beam line rotations and sextupoles.

B. Misalignments for Bending Magnets

The matrix giving the transformation of spatial coordinates from beginning to end of a bending magnet, as described above, were included in the program. However its terms were not consistent with the coordinate system conventions of TRANSPORT.

The direction convention for the bend angle used to calculate the transfer matrix R was opposite to that used to calculate the transformation of spatial coordinates. This error has now been corrected and sections containing bending magnets may now be misaligned.

C. Nested Misalignments

At times it is of interest to have one misaligned section contained within another. In such cases the cards specifying the misalignments (8. type code) may be placed together but pertain to different sections of the beam. The section to which the second card applies may begin somewhat earlier than that to which the first card applies.

A good example is found on page E-118 of the TRANSPORT manual[1], where the misalignment cards are as follows:

```
8.   0.1   0.1   0.1   0.1   0.1   0.1   100.   ;
8.   0.1   0.1   0.1   0.1   0.1   0.1   101.   ;
8.   0.1   0.1   0.1   0.1   0.1   0.1   102.   ;
```

In the old version of the program a misalignment updates the R_1 matrix. The second misalignment card in the example specifies that the beginning of the misaligned section is at the location of the last R_1 update, which is the previous card. Naively one would expect a null result, but from the example one sees that this does not happen. The reason is somewhat complex. When an update occurs of either R_1 or R_2, the transfer matrix is not set equal to the identity. The update is merely remembered so that when the next element having a transfer matrix is encountered its individual transfer matrix becomes the cumulative transfer matrix. The matrix giving the transformation of spatial coordinates is, however, immediately reset. This means that the result given by the example is meaningless.

The program has been changed such that all matrices are reset only when the next element with a transfer matrix is encountered. Calculations concerning nested misalignments may now be done.

D. Calculation of Misalignment Matrices

In the past, the matrices used for misalignment were always evaluated, whether one was doing a misalignment calculation or not. For example, they were calculated at each run through the beam line, when one was doing iterations in order to fit constraints. This was clearly a waste of computer line and very inefficient. The misalignment matrices are now calculated only when one is doing a misalignment calculation.

V. FITTING AND CONSTRAINTS

A. Suppression of Unwanted Output

When one is doing fitting with TRANSPORT, the program will make an initial run through the beam line and print out the sequence of elements along with transfer or beam matrices where specified. It then makes a series of runs through the beam line, adjusting the varied parameters to satisfy the constraints. Finally, it makes a final run through the beam line with print out, where the varied parameters are set at their final values.

When one is doing sequential fitting all the mentioned steps are taken at each stage of the fitting. Yet in this case the first run at a given stage is often almost identical to the second run at the previous stage, and is therefore not always desired.[4] Provision has therefore been made to allow the user to eliminate the first run through the beam line in the second and later stages of a sequential fit. If one places the number ⁻1 on the indicator card the first run through will not take place.

If one is not fitting, the program will make only one run through the beam line. However, the old version of TRANSPORT determined whether or not one was fitting by whether there were any vary codes present. If there were vary codes but not constraints, the program would make an initial run through the beam line, start to fit only to discover no unsatisfied constraints, change all varied parameters by zero, and make the final run. The program has been changed so that both vary codes and constraints are required for fitting. If either is missing the program will make only one run through the beam line.

The above described conditions leave a troublesome question. What if one is not fitting but by mistake places a ⁻1 on the indicator card to suppress the first, and now only, run through the beam line. The user is saved here. When the program sees a ⁻1 on the indicator card, it requires also that both vary codes and constraints be present or it will not suppress the first run through the beam line.

B. Varying the Length of a Bending Magnet

It is now possible to vary the length of a bending magnet to achieve a fit. One simply places a very code in the appropriate location on the card specifying a bending magnet.

C. Solenoids

According to the TRANSPORT manual, it is possible to vary both the length and the magnetic field of a solenoid. However, the partial derivatives of the transfer matrix with respect to the varied parameters were incorrect, so that, in fact, it was not possible, in general, to achieve a fit. The expressions for the partial derivatives mentioned have been corrected so that it is now possible to achieve a fit by varying the parameters of a solenoid.

D. Coupling of Varied Parameters and Limits

Limits are placed on the range of certain varied parameters by TRANSPORT. When, during a fitting procedure, a parameter goes outside the specified range it is reset to the end value of the range. If several parameters are coupled together (via vary codes 2-9) and one exceeds the allowed range of one, it will be reset while the others are not. Thus if one exceeds the internally set limits the coupling will be broken.[5,6]

An example of the above would be a pair of inversely coupled drift lengths. Such a procedure is useful if one wishes to vary the position of an element without affecting the remainder of the beam line. The lower limit placed on drift lengths by TRANSPORT is 0.01 of whatever measure of longitudinal length is being used. If two drift lengths are inversely coupled, and during fitting a change is made such that one of them would be less than this minimum, then that length is reset to 0.01 while the original change is made in the other. The position of all beam elements after the second mentioned drift thus changes. The undesirable nature of such a feature is clear.

In the new version of TRANSPORT all couplings are preserved when any limits are exceeded. The program calculates the changes to be made in all varied parameters. It then checks the tentative new value of each parameter agains the corresponding limit. If the parameter does not fall within the allowed range, a new value of the change to be made is calculated, such that the final value of the parameter in question will lie at the end point of the allowed range.

E. First Moment Constraints

It has always been possible to place a constraint on the beam centroid position. Such a constraint has been indicated by setting the first integer index on the constraint card equal to seven and the second to the number of the coordinate to be constrained. It was felt that the number seven was related to the centroid only via the internal workings of the program, and that setting this index equal to zero was a more natural indicator for the centroid. A fit of the beam centroid is now indicated when the first integer index is set equal to either seven or zero.

F. Constraints on Total Length

Often one will want to vary the lengths of certain elements while constraining the total length of the beam. Some of these lengths may be inversely coupled to others. In most cases, varied lengths are those of drift spaces, but there are times one might wish to vary the lengths of other elements.[7] For many of these other elements varying their length to fit the total beam length simply did not work. In other cases an inverse coupling was calculated as if obverse. All such cases have been corrected and the coupling of all lengths, varied inversely or obversely, to the total beam length is done correctly.

G. Accumulation of Partial Derivatives of Transfer Matrices

The method used when fitting to calculate the partial derivatives of the transfer matrices with respect to the varied parameters has been changed to achieve greater efficiency. Let us consider the accumulation of the R_2 matrix and its derivatives. In the past, when one traversed an element with an individual R matrix the cumulative R_2 matrix of the beam was calculated as:

$$R \times R_2 \rightarrow R_2.$$

The partial derivatives with respect to the varied parameters v_i are accumulated as:

$$R \times \frac{\partial(R2)}{\partial v_i} + \frac{\partial R}{\partial v_i} \times R_2 \rightarrow \frac{\partial(R2)}{\partial v_i},$$

where two derivative terms occur due to the possibility of coupling varied parameters. For an element with a transfer matrix but no varied parameters the matrices are accumulated as:

$$R \times R_2 \rightarrow R_2$$

$$R \times \frac{\partial(R2)}{\partial v_i} \rightarrow \frac{\partial(R2)}{\partial v_i} .$$

When one encounters a sequence of elements with no parameters being varied within the sequence, then the number of matrix multiplications at each element is one greater than the number of independently varied parameters previously occurring. Long sequences of elements where nothing is being varied are quite common both in NAL beams and in any beam where sequential fitting is being used.

In the part of the program which does the fitting, this procedure has been replaced by a more efficient one. In any section where no parameters are being varied a single matrix R_3 is carried. As one traverses an element it is accumulated as follows:

$$R \times R_3 \rightarrow R_3.$$

When a varied element is encountered, the R_2 matrix and its derivatives are calculated as:

$$R_3 \times R_2 \rightarrow R_2$$

$$R_3 \times \frac{\partial (R2)}{\partial v_i} \rightarrow \frac{\partial (R2)}{\partial v_i}.$$

Then one traverses the varied element using the procedure previously employed. Such a procedure substantially reduces the computer time used to run through a beam line when fitting.

H. Upper and Lower Limit Constraints

By placing a vary code on a card indicating a constraint, one can cause that constraint to serve as either an upper or a lower limit. Previously this vary code was ignored and the intended limit was taken as an ordinary constraint. The program has been changed so that now one can indeed specify a constraint as an upper or lower limit.

I. Variation of Elements of an Arbitrary Matrix

When one inserts an arbitrary matrix (type code 14.) into a beam line and varies one or more of the elements to obtain a fit, one often has little idea what the final value of that element will be. Therefore it would be perfectly reasonable to start with a value of zero. In the old version of the program this caused a division of zero by zero to be made, and therefore did not work properly. The possibility of such a spurious quotient has been eliminated and one can start with any estimate one deems appropriate.

J. Beam Correlation with Centroid Shift

It is possible to place a constraint on the beam correlation $r_{ij} = \sigma_{ij}/\sqrt{\sigma_{ii}\ \sigma_{jj}}$, where σ_{ij} indicates an element of the beam matrix. However the mechanism of fitting did not allow for the possibility of a centroid shift (type code 7.). Now one can fit correlations when a centroid shift is present.

VI. BEAM LAYOUT

One can now produce a coordinate layout of a beam line in any coordinate system one chooses. The coordinates printed are those of the beam line axis at the interface between any two elements. All three spatial coordinates are given.

A layout is specified by placing a 13. 12. ; card before the beam card. If no additional cards are inserted the beam will be assumed to start at the origin and proceed along the positive z axis. The y axis will point up and the x axis to the left. One can specify other starting coordinates and orientations by placing certain other cards also before the beam card as follows:

16. 16. x_o, y_o, and z_o respectively, the coordinates
16. 17. of the initial point of the beam line
16. 18.
16. 19. θ_o and ϕ_o the initial horizontal and vertical
16. 20. angles of the beam line.

When specifying the initial orientation of the beam line via the two angles, one must give the horizontal angle first. The meaning of the two angles is given in Figure 1.

VII. TYPICAL EXECUTION TIMES

For many problems, some of the above mentioned changes have resulted in an increase in efficiency. Below is a table giving execution times on the NAL PDP-10 of a few sample problems.

Problem	Old Program	New Program
Matrix Element Test Deck	11 m 42 s	1 m 7 s
SLAC Switchyard Fit Test	6 m 47 s	4 m 32 s
Second Order Run of NAL Beam M6	47 m 51 s	1 m 15 s

REFERENCES

1. Karl L. Brown, Sam K. Howry, SLAC Report No. 91 (1970).
2. Alan Wehmann, private communication.
3. J. Lamsa, private communication.
4. Joseph Lach, private communication.
5. Maurice Bourquin, private communication.
6. William Hermansfeldt, private communication.
7. John Schivell, private communication.

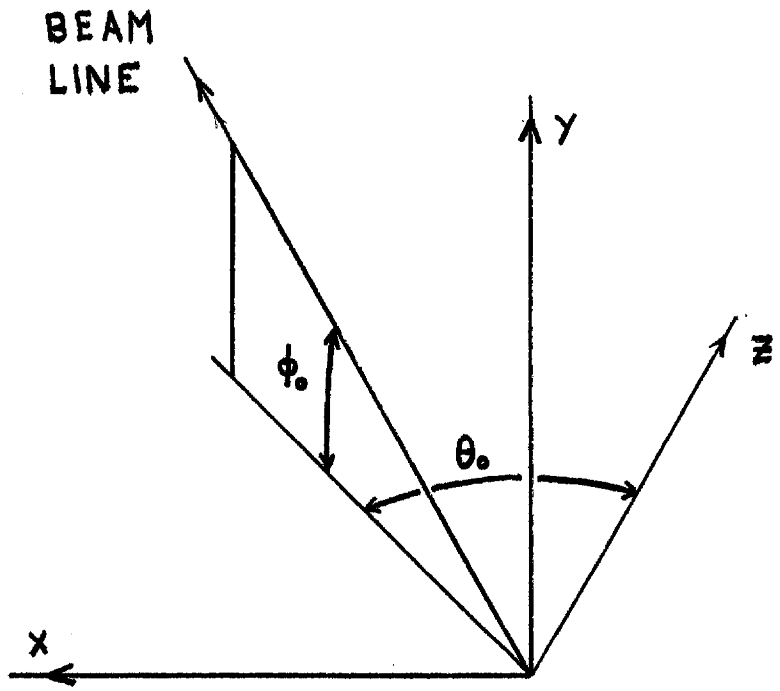

Fig. 1 Specification of initial angles θ_o and ϕ_o for beam line layout.